Collective Multi-Label Classification

Nadia Ghamrawi
University of Massachusetts Amherst
Amherst, Massachusetts, USA
ghamrawi@cs.umass.edu

Andrew McCallum
University of Massachusetts Amherst
Amherst, Massachusetts, USA
mccallum@cs.umass.edu

ABSTRACT

Common approaches to multi-label classification learn independent classifiers for each category, and employ ranking or thresholding schemes for classification. Because they do not exploit dependencies between labels, such techniques are only well-suited to problems in which categories are independent. However, in many domains labels are highly interdependent. This paper explores multi-label conditional random field (CRF) classification models that directly parameterize label co-occurrences in multi-label classification. Experiments show that the models outperform their single-label counterparts on standard text corpora. Even when multi-labels are sparse, the models improve subset classification error by as much as 40%.

Categories and Subject Descriptors

I.5.1 [**Pattern Recognition**]: Models—*statistical,structural*

General Terms

Design,Experimentation,Performance

Keywords

Classification, machine learning, multi-label, statistical learning, uncertainty

1. INTRODUCTION

Single-label classification assigns an object to exactly one class, when there are two or more classes. *Multi-label classification* is the task of assigning an object simultaneously to one *or multiple* classes.

The most common approach independently learns a binary classifier for each class, and then assigns to a test instance all of the class labels for which the corresponding classifier says "yes." Experiments have shown that the classifiers such as Widrow-Hoff, k-nearest-neighbor, neural networks and linear least squares fit mapping are viable techniques for this approach [17], as are support vector machines [8]. Although some binary classifiers provide posterior probability over their binary answers, they need only have binary valued output.

Another approach requires a real-valued score for each class, suitable for ranking class labels, and then classifies an object into the classes that rank above a threshold. Schapire Singer Schapire99 [14] develop a boosting algorithm that gives rise to such a ranking. The model described by Crammer Singer Crammer02 [5] learns a prototype feature vector for each class, and a class rank is derived from the angle between its prototype and the document. The model in Gao *et al.* Gao04 [6] trains independent classifiers for each category that may share some parameters, and ranks each classification according to a confidence measure.

The above methods learn independent classifiers for each class. However, it is often the case that there are strong co-occurrence patterns and dependencies among the class labels. Explicitly leveraging these patterns may be advantageous. For example, the belief that a research article having the word *sodium* is likely to be labeled HEART DISEASE supports the belief that the document should also be given the label HYPERTENSION. A method that captures dependencies between class labels is likely to provide improved classification performance, particularly for more richly multi-labeled corpora than those used in experiments.

This paper presents two multi-label graphical models for classification that parameterize label co-occurrences. As in traditional classifiers, both models learn parameters associated with feature-label pairs. The *Collective Multi-Label classifier* (CML) also, jointly, learns parameters for each pair of labels. The *Collective Multi-Label with Features classifier* (CMLF) learns parameters for feature-label-label triples—capturing the impact that an individual feature has on the co-occurrence probability of a pair of labels.

We present experiments using two data sets that, although sparsely multi-labeled, have become standard for multi-label classification experiments: the Reuters-21578 and OHSU-Med text corpora. CML and CMLF outperformed the binary models: they reduced error in subset accuracy by as much as 27%, reduced error in macro- and micro- averages by up to 9%, and had consistently better performance than their binary counterparts.

2. THREE MODELS FOR MULTI-LABEL CLASSIFICATION

Conditional probability models for classification offer a rich framework for parameterizing relationships between class labels and features, or characteristics, of objects. Furthermore, such models often outperform their generative counterparts.

CIKM'05, October 31–November 5, 2005, Bremen, Germany.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2005		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Collective Multi-Label Classification	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Air Force Office of Scientific Research,875 North Randolph Street Suite 325,Arlington,VA,22203-1768**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**7**	

Conditionally trained undirected graphical models, or conditional random fields (CRFs) [9], can naturally model arbitrary dependencies between features and labels, as well as among multiple labels. These dependencies are represented in the form of new (larger) cliques, which allow various clique parameterizations to express preferences for arbitrary types of co-occurrences.

Traditional maximum entropy classifiers, e.g. [13], are trivial CRFs in which there is one output random variable. We begin by describing this traditional classifier, then we describe its common extension to the multi-label case (with independently-trained binary classifiers), and then we present our two new models that represent dependencies among class labels.

2.1 Single-label Model

In single-label classification, any real-valued function $f_k(\mathbf{x}, y)$ of the object $\mathbf{x}$ and class y can be treated as a *feature*. For example, this may be the frequency of a word w_k in a text document, or a property of a region r_k of an image. Let V be a vocabulary of characteristics. The constraints are the expected values of these features, computed using training data. Suppose that $\mathcal{Y}$ is a set of classes and λ_k are parameters to be estimated, which correspond to features f_k, where k enumerates the following features:

$$k \in \{\langle v_i, y_j\rangle : 1 \le i \le |V|, 1 \le j \le |\mathcal{Y}|\}.$$

That is, k is an index over features, and each feature corresponds to a pair consisting of a label and a characteristic (such as a word). Then the learned distribution $p(Y|\mathbf{x})$ is of the parametric exponential form [1]:

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_k \lambda_k f_k(\mathbf{x}, y)\right), \quad (1)$$

$Z(\mathbf{x})$ is the normalizing factor over the labels:

$$Z(\mathbf{x}) = \sum_y \exp\left(\sum_k \lambda_k f_k(\mathbf{x}, y)\right). \quad (2)$$

Given training data

$$D = \{\langle \mathbf{x_1}, y_1\rangle, \langle \mathbf{x_2}, y_2\rangle, ..., \langle \mathbf{x_r}, y_r\rangle\},$$

the penalized log likelihood of parameters Λ is

$$l(\Lambda|D) = \log\left(\prod_{d=1}^{r} p(y_d|\mathbf{x_d})\right) - \sum_k \frac{\lambda_k^2}{2\sigma^2}$$
$$= \sum_{d=1}^{r}\sum_k (\lambda_k f_k(\mathbf{x_d}, y_d) - \log Z_\Lambda(X)) - \sum_k \frac{\lambda_k^2}{2\sigma^2}, \quad (3)$$

where the last term is due to the Gaussian prior used to reduce overfitting. The trainer attempts to find a Λ that maximizes $l(\Lambda|D)$ iteratively. The gradient of the log likelihood at k is

$$\frac{\delta(\Lambda|D)}{\delta\lambda_k} = \sum_{d=1}^{r}\left(f_k(\mathbf{x_d}, y_d) - \sum_y f_k(\mathbf{x_d}, y)p(y|\mathbf{x_d})\right) - \frac{\lambda_k}{\sigma^2}. \quad (4)$$

Since this cannot be solved analytically in closed form, the optimal λ is found by convex optimization. BFGS [3] is a fast optimization method that finds the global maximum of the likelihood function given the value and gradient.

2.2 Accounting for Multiple Labels

The single-label model above learns a distribution over labels. In a multi-label task, the model should learn a distribution over *subsets* of the set of labels $\mathcal{Y}$, which are represented as bit vectors $\mathbf{y}$ of length $|\mathcal{Y}|$.

In the most general form, given instance $\mathbf{x}$ and features f_k,

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_k \lambda_k f_k(\mathbf{x}, \mathbf{y})\right), \quad (5)$$

where $Z(\mathbf{x})$ is the normalizing constant. All three CRF models capture the following enumeration over features in the learned distribution:

$$k \in \{\langle v_i, y_j\rangle : 1 \le i \le |V|, 1 \le j \le |\mathcal{Y}|\}.$$

That is, all three models capture the dependency between each object feature and each label.

2.2.1 Binary Model

A common way to perform multi-label classification is with a binary classifier for each class. For each label y_b, the binary model trains an independent binary classifier c_b, partitioning training instances into positive (+) and negative (−) classes (Figure **??**). The learned distribution p_b is as in Equation **??**, except that

$$k \in \{\langle v_i, r_j\rangle : 1 \le i \le |V|, 0 \le j \le 1\}$$

since $r_j \in \{+, -\}$. However, the distribution over multi-labelings, $p(\mathbf{y}|\mathbf{x})$ is as follows:

$$p(\mathbf{y}|\mathbf{x}) = \prod_b p_b(y_b|\mathbf{x}). \quad (6)$$

This scheme attributes an object $\mathbf{x}$ to category labeled y_b if c_b classifies $\mathbf{x}$ positively. However,the classifications are treated independently.

Figure **??** depicts this model as a factor graph. The black squares (factors) represent the model parameters. For example, in Figure 1(a), the binary model maintains a parameter for each pair consisting of a label and a feature. Factor graphs are graphical models that depict the clique parameterizations. Inference in factor graphs is done in a way similar to inference in graphical models [10].

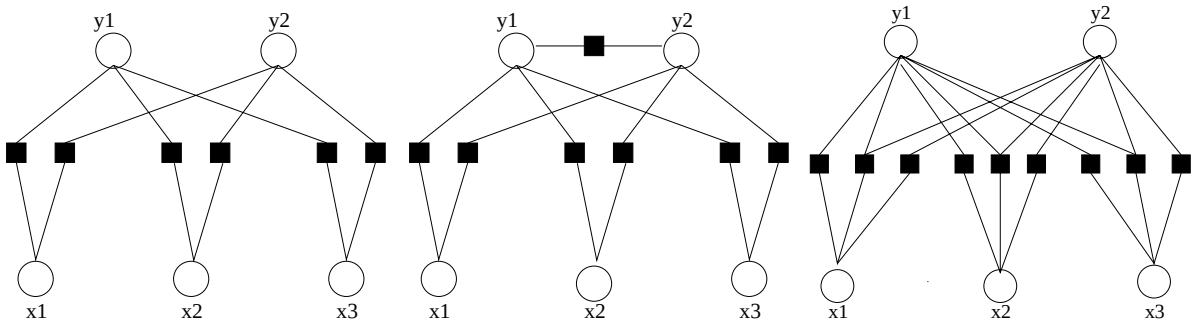
2.2.2 CML Model

In order to capture co-occurrence patterns among labels, this paper presents a conditional random field representing dependencies among the output variables.

In addition to having feature for each label-term pair, CML maintains features accounting for label co-occurrences. This model is depicted in Figure **??**. For object $\mathbf{e}$ and labels y' and y", there are four features:

	feature
0	neither y' nor y" labels $\mathbf{e}$
1	y' but not y" labels $\mathbf{e}$
2	y" but not y' labels $\mathbf{e}$
3	both y' and y" label $\mathbf{e}$

For $k' = \langle$WHEAT, GRAIN, $2\rangle$ and training document $\langle \mathbf{x}, \mathbf{y}\rangle$, $f_{k'}(\mathbf{x}, \mathbf{y})$ is 1 if $\langle \mathbf{x}, \mathbf{y}\rangle$ is labeled GRAIN but not WHEAT, and 0 otherwise. A document has $4\binom{|Y|}{2}$ such features.



(a) Single-label model. (b) CML model. (c) CMLF model.

Figure 1: Factor graphs representing the multi-label models, where y_i is a label and x_i is a feature, and the black squares represent clique parameterizations. In (a) each parameterization involves one label and one feature. Figure (b) represents an additional parameterization involving pairs of labels, and figure (c) represents a parameterization for each label and each feature, together with each pair of labels and each feature.

The distribution $p(\mathbf{y}|\mathbf{x})$ thus becomes

$$\frac{1}{Z_\Lambda(\mathbf{x})} \exp\left(\sum_k \lambda_k f_k(\mathbf{x},\mathbf{y}) + \sum_{k'} \lambda_{k'} f_{k'}(\mathbf{y})\right) \quad (7)$$

where $Z_\Lambda(\mathbf{x})$ is the normalizing constant and

$$k \in \{\langle v_i, y_j\rangle : 1 \le i \le |V|, 1 \le j \le |\mathcal{Y}|\},$$

$$k' \in \{\langle y_i, y_j, q\rangle : q \in \{0,1,2,3\}, 1 \le i,j \le |\mathcal{Y}|\}.$$

The log likelihood $l(\Lambda|D)$ is similar to Equation **??**:

$$\sum_{d=1}^{r}\left(\sum_k \lambda_k f_k(\mathbf{x_d},\mathbf{y_d}) + \sum_{k'} \lambda_{k'} f_{k'}(\mathbf{y_d}) - \log Z_\Lambda(\mathbf{x_d})\right) - \sum_k \frac{\lambda_k^2}{2\sigma^2} - \sum_{k'} \frac{\lambda_{k'}^2}{2\sigma^2}. \quad (8)$$

The computation of the gradient is analogous to Equation **??**. CML captures the label co-occurrences in the corpus independent of the object's feature values. Effectively, for each label set, it adds a bias that varies proportionally to the label set frequency in training data. The factor graph for this model is depicted in Figure **??**.

2.2.3 CMLF Model

While CML parameterizes the dependencies between labels in general, these dependencies do not account for the presence of particular observational features (e.g., words). The tendency of labels to occur together in a multi-labeling is not independent of the appearance of the observational features. For instance, a text document belonging to the categories RICE and SOYBEAN might have increased likelihood of being correctly classified if the document has the word *cooking*, but decreased likelihood of belonging to ALTERNATIVE FUELS. The factor graph in Figure **??** reflects this dependency. The CMLF model maintains parameters that correspond to features for each $\langle term, label_1, label_2\rangle$ triplet, capturing parameter values for $\langle cooking, \text{RICE}, \text{SOYBEAN}\rangle$, for example.

As with CML, CMLF defines feature parameters over the labels and words,

$$k \in \{\langle v_i, y_j\rangle : 1 \le i \le |V|, 1 \le j \le |\mathcal{Y}|\},$$

but also defines parameters over pairs of labels and words,

$$k' \in \{\langle v_i, y_j, y_{j'}\rangle : 1 \le i \le |V|, 1 \le j, j' \le |\mathcal{Y}|\},$$

for a total of $O(n^2|V|)$ parameters for n labels. Note that CMLF maintains overlap in term occurrences: it has a feature for each pair consisting of a term and a label, as well as a feature for each triplet consisting of a term and two labels. The features enumerated by k provide some shrinkage, and thus protection from overfitting [4].

The corresponding distribution that CMLF learns is

$$\frac{1}{Z_\Lambda(\mathbf{x})} \exp\left(\sum_k \lambda_k f_k(\mathbf{x},\mathbf{y}) + \sum_{k'} \lambda_{k'} f_{k'}(\mathbf{x},\mathbf{y})\right). \quad (9)$$

The gradients of the log likelihood at k and at k' are the same as those of CML, except that k' enumerates different features. CML has four features for each pair of labels, while CMLF has $|V|$ features. The factor graph for this model is depicted in Figure **??**; note that for each observational feature, there is a parameter for each label, and also a parameter for each pair of labels.

Parameter estimation in these models is the same as for the single-label model: calculation of the value and gradient is straight-forward, and BFGS is used to find the optimal parameters given the gradient of the log-likelihood. Note that neither multi-label model assumes that the label taxonomy has a complex structure, although extra parameters accounting for this could easily be added.

Table **??** shows the asymptotic complexity of training an instance. The binary technique is faster than the multi-label models in most cases, but performance of binary pruning depends on selection of the threshold, which determines the number of classes. In large datasets with many rarely occurring multi-labelings, binary pruning requires considerably less training time than supported inference, for comparable classification performance. Experiments suggest that the binary pruned inference technique is faster than supported inference. CMLF is linear with respect to CML, which is asymptotically simpler than the binary classifier method only if the multi-labelings are sparse. However, in practice binary classifiers are faster to train because they use fewer parameters in optimization.

	binary	CML	CMLF
supported	k^2v	$s(av + k^2)$	sa^2v
pruned	k^2v	$2^r(rv + k^2)$	$2^r r^2 v$

Table 1: Asymptotic per-instance training complexity, given $|V| = v$, k labels, s total label combinations of average size a and r labels ranking above threshold on average.

3. INFERENCE

Rather than providing a probability estimate for each label, exact inference using the collective models requires learning a probability distribution over all possible multi-labelings — that is, over all subsets of $\mathcal{Y}$. This method is intuitively appealing: it is easy to explain, and it is informative, since it offers a probability score for each combination of labels, regardless of the combination presence in the training data. However, since the number of subsets is exponential in the number of class labels, the problem is tractable only for about 3-12 classes. When the number of classes is larger, approximate inference methods may prune certain combinations of labels, and calculate the conditional distribution over the pruned set.

One method of pruning is to include only the label combinations that that occur in training data—which we term the *supported* combinations. This method can sometimes be surprisingly effective. For the top 10 classes in Reuters-21578, only 0.6% of test instances belong to combinations of categories that do not occur in training data. For the entire *ModApte* split, the error due to supported inference is more significant: 4% of test instances have label combinations that do not occur in training data. When there are few classes and few such outliers, or when such rare combinations can be excluded, then supported inference is a very good solution.

An alternative approximate inference method is termed *binary pruned inference*, and represents a compromise between supported and exact inference. The model trains an independent binary classifier for each label. Then when classifying an object, exact inference considers only the labels having binary classifier probability scores above a certain threshold (t). Cross validation on training data is used to choose the threshold.

Binary pruned inference makes it possible to correctly classify test documents whose actual combinations do not occur in the training data. Furthermore, the method requires less training time than supported inference.

4. EXPERIMENTS

We present experiments with these multi-label classifiers on two standard multi-label data sets: Reuters-21578 and the 'Heart Disease' (*HD*) documents of OHSU-Med. The corpora differ in the noise level and length of documents. Both have simple label taxonomies: labels are not hierarchical, and each document has at least one label from the entire label set.

Except in the case of the k' features of CML, features f_i are represented by count of occurrences, in experiments presented here. Alternate representations include frequency of occurrences, for example.

[2]The mis-classification rate is the percent of times that the binary classifier incorrectly assigns one of the labels to an object, or fails to assign the correct label to an object.

4.1 Corpora

The *ModApte* split of Reuters-21578, in which all labeled documents that occur before April 8, 1987 are used in training and other labeled documents are used in testing, is a popular benchmark for experiments. The *ModApte* documents consist of those documents labeled by the 90 classes which have at least one training and one testing instance, accounting for 94% of the corpus. Roughly 8.7% of these documents have multiple topic labels.

Experiments using corpus *Reuters10* use only documents belonging to the 10 largest classes, which label 84% of the documents and form 39 distinct combinations of labels in the training data. Table **??** depicts the distribution of multi-label cardinalities in the *ReutersAll* test set, together with the label classification error rate of the binary classifiers.

The OHSU-Med [7] *HD* corpus, a popular dataset for text classification, is a collection of titles and abstracts of medical research journal articles from 1989-1991 corresponding to characterizations of the relevant heart conditions, such as "Heart Aneyurism" and "Myocarditis". The *HD-small* documents belong to the 40 categories which label between 15 and 74 training documents, forming 106 combinations of labels in the training data. *HD-big* consists of documents belonging to the remaining 16 categories that each label 75 or more training documents.

4.2 Results

Features are ranked according to their mutual information, so that the classifiers may select a proportion of features having the highest rank. Parameters that influence performance of the classifiers include proportion of features selected, Gaussian prior variance of the parameters, and in the case of binary pruning, the threshold for the binary classifiers. The classifiers are least sensitive to the Gaussian prior, and binary pruning is most sensitive to the threshold. Lower thresholds have higher classification cost but higher thresholds limit the performance of CML and CMLF to the performance of the binary classifiers.

In experiments presented in this paper, words occurring fewer than 5 times in all training documents are excluded from the vocabulary, and all classifiers assume a Gaussian prior variance of 1.0. Thresholds and feature proportions are learned using cross validation on training data. That is, the parameters that a given classifier uses are those which yield the best average performance, of the binary model and its multi-label counterpart, using a random partition of the training data into training and validation instances.

The results are compared using three metrics: F1 micro-average, F1 macro-average [17], and subset accuracy. The macro-average is the mean of the F1-scores of all the labels, thus attributing equal weights to each F1-score. The micro-average is the F1-score obtained from the summation of contingency matrices for all binary classifiers. The micro-average metric gives equal weight to all classifications, so that F1 scores of larger classes influence the metric more than F1 scores of smaller classes. F1-score reflects the harmonic mean of precision and recall. Subset accuracy is the proportion of documents with entirely correct bit vectors $\mathbf{y}$.

4.2.1 Reuters-21578

Even for the sparsely multi-labeled *ReutersAll*, CMLF reduces error in F1 averages by as much as 5%, and reduces error in subset classification by 16%. Table **??** depicts the results of experiments on *ReutersAll* using the *ModApte* split, as well as a comparison of

number of labels	1	2	3	4	5	6	7-14
number of documents	2561	308	64	32	14	6	13
binary model error	0.142%	0.641%	1.46%	1.98%	1.85%	3.33%	5.83%

Table 2: Histogram of *ReutersAll* test set combinations of labels by combination cardinality, and the binary model label misclassification rate.[2] As the cardinality of an object's multi-labeling increases, the binary models are more likely to incorrectly an individual label. This trend suggests that it is advantageous to leverage label co-occurrences in classifying documents.

ReutersAll, ModApte				
	Binary	CML	Binary	CMLF
Supported				
	40% words		50% words	
macro-F1	0.4380	**0.4478**	0.4380	**0.4477**
micro-F1	0.8627	**0.8659**	0.8627	**0.8635**
sub. acc.	0.7999	**0.8329**	0.7999	**0.8316**
cl. time (ms)	1.4	48	1.4	78
Binary pruned				
	70% words,$t = 0.3$		50% words,$t = 0.4$	
macro-F1	0.4384	**0.4792**	0.4388	**0.4760**
micro-F1	0.8629	**0.8692**	0.8634	**0.8701**
sub. acc.	0.8000	**0.8119**	0.8000	**0.8162**
cl. time (ms)	1.4	4.6	1.4	4.7

Table 3: Performance of the three inference techniques. Feature proportions, and threshold parameters for binary pruning (t), are learned using cross-validation on training data. Even for this sparsely multi-labeled corpus, the multi-label models always outperform their binary counterparts, reducing error in subset accuracy by as much as 8% and in F1 scores by 5-8%.

HD-small				
	Binary	CML	Binary	CMLF
Supported				
	70% words		70% words	
macro-F1	0.5846	**0.6224**	0.5846	**0.6200**
micro-F1	0.6138	**0.6426**	0.6138	**0.6440**
sub. acc.	0.4096	**0.5489**	0.4096	**0.5721**
Binary Pruned				
	70% words,t=0.9		70% words,t=0.4	
macro-F1	0.5846	**0.6038**	0.5846	**0.6028**
micro-F1	0.6138	**0.6189**	0.6138	**0.6158**
sub. acc.	0.4096	**0.4818**	0.4096	**0.4634**
HD-big				
	Binary	CML	Binary	CMLF
Supported				
	70% words		70% words	
macro-F1	0.6467	**0.6795**	0.6483	**0.6629**
micro-F1	0.6834	**0.7003**	0.6849	**0.6983**
sub. acc.	0.4914	**0.5925**	0.4914	**0.6025**
Binary Pruned				
	70% words,t=0.6		70% words,t=0.3	
macro-F1	0.64676	**0.6556**	0.6482	**0.6658**
micro-F1	**0.6839**	0.6751	0.6849	**0.6886**
sub. acc.	0.4910	**0.5226**	0.4918	**0.5190**

Table 4: Results of experiments on *HD*, trained on documents from 1991 and tested on documents from 1990. Multi-label models reduce F1 macro and micro-average error by 8%.

the two inference methods. Supported inference experiments are more costly in time and space than binary pruning.

With *ReutersAll*, binary pruning generally performs better than supported inference. Furthermore CML and CMLF perform better than the best reported results.

The binary pruning technique resulted in 3% higher F1 micro-average and 23% higher macro-average than supported inference. The significant gain in macro-average suggests that binary pruning improves performance of smaller classes.

Collective classifiers perform better than the traditional binary model, supporting our contention that the classes are not independent, and that directly parameterizing these dependencies is advantageous.

4.2.2 OHSU-Med

HD is a noisier corpus than Reuters-21578, having topics that span a narrower semantic scope. As with Reuters-21578, CML and CMLF trump the traditional binary models. With thresholds chosen using cross validation on training data, CML and CMLF achieve better performance with supported inference than binary pruning.

In *HD*, typically more than half of the misclassifications in binary pruning are due to the pruning of positive classes. Thus on pruned instances, the F1 averages that the collective models achieve with supported inference are higher than the averages achieved using binary pruning.

Table **??** depicts performance of the five techniques on *HD-small* and *HD-big*. Compared to the traditional binary model, using supported inference, the collective classifiers improve subset accuracy by 20-40%, whereas with *ReutersAll*, this improvement is about 4%. (The collective models increase F1 averages by 5-9% for both *HD* corpora.) It is gratifying to see that on tasks with larger, more complex multi-labeled sets, our method provides even greater improvement.

The average improvement of CML and CMLF over binary classifiers is even greater across several trials using random test-train splits (of comparable proportions to those of Table **??** experiments) of the corpus. Experiments suggest that more innovative binary pruning models could improve performance considerably.

5. RELATED WORK

Some existing models indirectly leverage the multi-label dependencies that traditional methods do not. semantic scene classification, Boutell *et al.* Boutell03 [2] train a single-label classifier for each label, using all single-label documents and only the multi-label documents with that label. This approach indirectly leverages label co-occurrences, but it does not directly parameterize multi-label dependencies.

Expectation Maximization has been used to train a mixture model [11] for which the features of each document are produced by a mixture of word distributions for each class. [16] take a similar

approach in that each word in each category is generated from a multinomial distribution over vocabulary words. Both of these approaches are generative, and both leverage information about multiple class memberships for a given document implicitly by learning which classes generate which features.

Relational Markov Network models (RMNs) [15], are undirected graphical models like CML and CMLF. However, they perform single-label classification simultaneously of multiple documents, whereas CML and CMLF address the issue of multi-label classification of a single document. Furthermore, RMNs use the hyperlinks linking separate documents to capture dependencies between documents, but the model relies on the inherent sparseness of those dependencies, while CML and CMLF prove advantageous for densely multi-labeled corpora. RMNs use loopy belief propagation is used for estimating the gradient.

6. CONCLUSIONS AND FUTURE WORK

Multi-label classification is an important task in domains beyond text. In many real-world tasks, classes are not independent. CML and CMLF offer a framework for leveraging the dependencies between categories by including factors that capture label co-occurrences, whereas previous methods leverage category dependencies only indirectly, at best.

The success of conventional classification approaches depends on properties such as independence of classes and sparsity of multi-labelings. On varying corpora, over several metrics, the collective models outperform these methods.

Research related to multi-label classification involves automatically annotating biomedical abstracts with lists of genes that are mentioned in the documents. This is related to multi-label classification because each gene may have several synonyms, and a synonym may refer to several genes. More generally, in any domain in which subsets of unstructured interdependent outcomes are to be assigned, the CML and CMLF framework suggests a viable solution.

Future experiments may test the models in different domains and use corpora with varying noise characteristics, as well as domains in which features do not have uniform weight and type, including semantic scene classification.

Improved inference and pruning methods may be more tractable than exact and supported inference and allow greater flexibility than binary pruning.

A more general extension of CML and CMLF would parameterize larger factors, rather than pairs of labels, and incorporate schemes for learning which factors to include [12]. Enhanced models could also handle unlabeled data.

7. ACKNOWLEDGEMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by Air Force Office of Scientific Research contract #FA9550-04-C-0053 through subcontract #0250-1181 from Aptima, Inc., in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249, and in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] M. Boutell, X. Shen, J. Luo, and C. Brown. Multi-label semantic scene classification, technical report, dept. comp. sci. u. rochester. 2003.

[3] R. H. Byrd, J. Nocedal, and R. B. Schnabel. Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming*, 63:129–156, 1994.

[4] S. F. Chen and R. Rosenfeld. A gaussian prior for smoothing maximum entropy models, technical report cmucs -99-108, carnegie mellon university. 1999.

[5] K. Crammer and Y. Singer. A new family of online algorithms for category ranking. In *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 151–158, Tampere, Finland, 2002. ACM.

[6] S. Gao, W. Wu, C.-H. Lee, and T.-S. Chua. A mfom learning approach to robust multiclass multi-label text categorization. In *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004)*, Banff, Alberta, Canada, 2004. ACM.

[7] W. R. Hersh, C. Buckley, T. J. Leone, and D. H. Hickam. Ohsumed: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 192–201, Dublin, Ireland, July 1994. ACM/Springer.

[8] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Machine Learning: ECML-98, 10th European Conference on Machine Learning*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer, April 1998.

[9] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 282–289, Williamstown, MA, USA, 2001. Morgan Kaufmann.

[10] H. A. Loeliger. An introduction to factor graphs. In *IEEE Signal Processing Magazine*, pages 28–41, January, 2004.

[11] A. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI'99 Workshop on Text Learning*, 1999.

[12] A. McCallum. Efficiently inducing features of conditional random fields. In *UAI'03, Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 403–410, Acapulco, Mexico, August 2003. Morgan Kaufmann.

[13] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *In IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[14] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[15] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI '02, Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence, University of Alberta, Edmonton, Alberta, Canada, August 1-4, 2002*, pages 485–492, Edmonton, Alberta, Canada, August 2002. Morgan Kaufmann.

[16] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002*, pages 721–728, Vancouver, British Columbia, Canada, December 2002. MIT Press.

[17] Y. Yang. An evaluation of statistical approaches to text categorization. *Inf. Retr.*, 1(1-2):69–90, 1999.